

**SYSTEM AND METHOD FOR GENERATING
SOFTWARE TOOLS**

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to software tools and more specifically to Wizards.

Background Art

A software tool is a computer program that helps users
5 perform specific tasks. One such known software tool is a Wizard. Software tools, such as Wizards, aid a user in creating various documents or help a user through complex procedures.

Examples of known Wizards include setup Wizards for
10 the installation of software and document creation Wizards within word processing programs, which methodically step a user through the creation of documents. These known Wizards are not easily modifiable and are programmed in code. Wizards hide the complexity of the program from the
15 user, allowing the user to quickly achieve results that would otherwise require much training and skill. Since the programs hide complexity from the users, the programs are often complex themselves. Accordingly, an easier way to create and maintain Wizards, and to modify Wizards, is
20 desired.

An example of a Wizard for creating a facsimile coversheet is shown in Fig. 1 as element 10. The Wizard generally displays a sequence block 10a to a user, illustrating each of the steps necessary to create the
25 document, and displays a block 10b to a user, specifically dedicated to a particular step. As a step in the procedure

10a is selected, display 10b changes to correspond to the specifics of the selected step.

Initially, as shown in Fig. 1, a start step 2 begins the procedure. Thereafter, a user selects a next step 4, relating to a document to fax. Once this step 4 is selected, display portion 10b then changes to display a first question 20, and predetermined selectable answers 22. This question 20 is a first one which must be answered when creating a facsimile document, and the possible answers 22 allow for various predetermined variations of the document to be created. Similarly, a next question 2 may be posed as indicated by element 24, with its predetermined set of possible answers 26.

Thereafter, a user selects a next step 6 entitled recipient, and new questions and predetermined possible answers are then displayed in display portion 10b corresponding to the recipient portion of the facsimile document. Step 8 then provides for various types of coversheets to be created, based in various types of fonts for example, and step 12 thereafter allows for modifications of the document relating to the sender, wherein a default name could be entered for example. Finally, upon reaching step 14, the document is completed.

By using the Wizard as shown in Fig. 1, a fax coversheet can be quickly created. However, as previously stated, this fax Wizard has been preprogrammed with perhaps a complex program hidden from the person creating the facsimile coversheet document. It includes only predetermined questions and predetermined selectable answers and therefore is not easily modifiable by a user. In other words, although a user can create a somewhat customized facsimile sheet, the customization cannot in any way be altered by the user. Accordingly, a need for such modifiability exists in the creation of software tools,

such as software Wizards.

Additionally, many tasks do not have predesigned Wizards to simplify them. Further, many manuals involve complex inter-dependent procedures which need to be
5 simplified in some way to be more easily understood. Accordingly, a need exists for Wizard-type software tools in many environments, including the simplification of procedural manuals.

SUMMARY OF THE INVENTION

10 The present invention is directed to a method and apparatus for creating a customizable software tool. In essence, a software tool or Wizard builder is created which prompts input of process or procedural steps and a plurality of selections or answers associated with the
15 steps, stores the input process steps and associated potential selections, and creates a software tool based upon the stored process steps and associated potential selections. As such, a software tool can be created which is easily creatable to suit any environment or problem, and
20 which is further modifiable by the addition or subtraction of process steps and/or potential selections.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will become more fully understood from the detailed description given hereinbelow
25 and the accompanying drawings which are given by way of illustration only and are not intended to limit the invention, wherein like reference numerals represent like elements and wherein:

Fig. 1 illustrates an example of a known Wizard;

30 Fig. 2 illustrates a hardware diagram used for implementing the present invention;

Fig. 3 is a software tool creator of the present invention;

Figs. 3a and 3b illustrate additional aspects of the

software tool creator of the present invention;

Fig. 4 is a sample screenshot of a Wizard created by the method and apparatus of the present invention;

Fig. 4a is another sample screenshot;

5 Fig. 5 illustrates a state table;

Fig. 6 illustrates a state transition table;

Fig. 7 is a flowchart of a process to be implemented into a software tool;

10 Figs. 8 to 11 are sample screenshots for the software tool created to implement the procedure of the flowchart of Fig. 7;

Fig. 12 is a flowchart of a second process to be implemented into a software tool;

15 Figs. 13, 14, 16, 17 and 18 are sample screenshots of a software tool created by the method and apparatus of the present invention to implement the procedure of the flowchart of Fig. 12;

20 Fig. 15 is a sample screenshot of the software tool creator used to create the software tool screenshot of Fig. 14;

Fig. 19 is an article of manufacture implementing the present invention; and

Fig. 20 is a propagated signal implementing the present invention.

25 DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

The present invention is directed to a method and apparatus for creating a software tool, such as a Wizard, to enable a user to easily create a customized tool for performing a specific task, without the need of complex programming, and to enable a user to create a tool which is
30 easily modifiable to meet specific needs. Such a created tool can be one used to simplify a complex document, or to methodically step a user through the creation of an often used document, for example. Thus, unlike known Wizards

which are complex to program and which are preprogrammed and non-modifiable, the software tool creation method and apparatus of the present invention enables one to easily create new software tools for any procedure or situation, 5 and further enables the creation of tools which are easily modifiable to provide for the utmost flexibility.

The software tool creation method and apparatus of the present invention is preferably implemented in an apparatus 30, for example, including a CPU 32 operably connected to 10 a memory 34 and an interface 36, and optionally connected, through a second machine interface 38, to any external machine.

The memory 34 stores the software tool creating method or program, the contents of which will be described in more 15 detail with regard to the software tool creator or Wizard builder of Fig. 3. The memory 34 can optionally store a relational data base from which a software tool can be created, including a state table such as shown in Fig. 5 and a state transition table shown in Fig. 6.

20 A state is essentially a step in a procedure, with a state table being a table of a plurality of states in a procedure and corresponding attributes for each state. A state transition table is one that defines movements from state to state in response to a given input. Thus, in 25 response to an answer selected, the state transition table defines a next-state in a procedure, and additional attributes. The relational data base, including such state tables and state transition tables, can be one as shown in, for example, either one or both of copending applications 30 serial numbers 09/182,637 and 09/182,493, respectively entitled "Method and Apparatus for Storing Data as Liquid Information" and "Method and Apparatus for Amplifying Design Information into Software Products", each attributable to the inventor of the present application and

each incorporated herein by reference.

The interface 36 can be in the form of an integrated input and display apparatus such as a touch screen, for example, or can include separate input and output
5 interfaces such as a display (output) and a keyboard and/or mouse (input). Alternatively, or in addition, the inputting and outputting of information can be done audibly through a microphone (input) and speaker (output) wherein questions and choices of a created software tool, or
10 prompts for creating a software tool, are audibly output to a user. Accordingly, the interface for inputting and outputting information to and from the CPU 32 can encompass known input and output devices.

Machine interface 38 provides access to an external
15 machine. Thus, the software tool creator of the present invention can implement a software tool including process steps requiring information from an external machine, such as a computer. In the created software tool, the external machine may be designated to provide information in various
20 steps of the created software tool.

For example, a software tool can be created by the apparatus and method of the present invention, which requires data from an external machine. This can be implemented in a process step. In creating the process
25 step, the step is designatable as either a human type or machine type, with the machine type requiring use of a machine interface 38.

A software tool which has been created to implement a machine step may display something to a user during that
30 step, but will not require a user selection. Instead, the machine interface 38 will contact an external machine, provide the external machine with a request for information, and await receipt of the information from the machine. Upon receiving such information, the created

software tool then proceeds to the next step, either requiring more machine information or requiring input from a user. Such a machine interface 38, for example, can include a URL address, accessing a particular computer website on the worldwide web, for example, and thus requesting receipt of information external to the apparatus 30.

Fig. 3 illustrates an apparatus for creating a software tool, such as a Wizard, which has been designated Wizard builder 100. The Wizard builder 100 shown in Fig. 3 is implemented in the form of a display requesting input of information, through a keyboard for example, to create a software tool. Fig. 4 illustrates a screenshot corresponding to the information input to Wizard builder 100 of Fig. 3, the screenshot being a screen of a software tool created by the Wizard builder 100 of Fig. 3.

Wizard builder 100 of Fig. 3 includes a first predetermined section 102, corresponding to "state". A "state" corresponds to a step in a process. Accordingly, a user is prompted to input a process step in a first window 104. In the example shown in Fig. 3, the state "2", corresponding to second step in a process, is input into the window 104.

Next, "procedure" 106 is displayed by the Wizard builder 100, prompting input of the procedure title or process to be implemented by the Wizard builder 100, in a second window 108. This procedure 106 will not normally change as the various states change (although a software tool can be created which takes a user through multiple linked procedures), and is an indication of what is being implemented by the Wizard builder 100, namely what is the title of the software tool.

The example software tool shown in Fig. 4 is for reducing a complex manual to a simple procedure, wherein

the manual or procedure is entitled TCP-110. As shown in Fig. 4 for example, both the procedure 250 and the step or state 260 are continuously displayed to a user when the created Wizard is displayed. Upon moving to step 3 in the procedure, step 3 will be displayed in the area 260, and procedure TCP-110 will remain in area 250.

As shown in Fig. 3, the Wizard builder 100 preferably additionally displays the type of step or state 110 to be implemented, namely a human step or a machine step. Accordingly, a user is prompted to input either an "M" for a machine step or an "H" for a human step in a third window 112. For a human step, the Wizard builder 100 will thereafter prompt input of process steps and potential selections associated with the process steps, wherein for a machine step, the machine address and information relating to the machine processing will be prompted. In the eventually created software tool, while human steps will require selection of a displayed potential selection, the machine step may be displayed (it may also optionally be hidden from a user), but information will be requested from the machine, and no potential selections from a user will be required.

A starting state window 114 and an ending state window 116 are also displayed by Wizard builder 100, prompting a user to indicate when a certain step is a starting state (first step) or an ending state (last step) in a procedure.

Element 120 is a "new state" button which can be accessed via a mouse or touch-screen for example, as can "apply state" button 122 and "delete state" button 124. Once information corresponding to the state 102, procedure 106, and question or procedural step 140 is input by the user, a state is then applied and the "apply state" button 122 is selected. If a state is no longer desired, the "delete state" 124 button is selected. If an additional

state or step in a process is desired, the "new state" 120 button is selected. Thus, any created software tool is easily modifiable by the insertion of new steps or deletion of unwanted steps.

5 Display area 130 of Wizard builder 100 shown in Fig. 3, preferably displays each of a plurality of steps in the process or procedure, highlighting the current state or step in area 132, such as state 2 in the present example.

Wizard builder 100 further displays "question" 140,
10 prompting input of a question or procedural step to be carried out in another window 142. Thus, the question corresponding to the second state or second step in a process, such as "is configuration simplex or duplex?" in the current example, is input in window 142..

15 As previously stated, a state can be a procedural step for a human to execute (or a question for a human to answer), or a machine state. With regard to machine states, if the machine state requires access to a computer, for example, a computer address may be necessary.
20 Accordingly, the Wizard builder 100 includes a prompt 144, prompting input of a URL address into a window 145. Since the present example shown in Fig. 3 is directed to a human step, not a machine step, the URL address window 145 is blank.

25 Another section of Wizard builder 100 shown in Fig. 3 involves the prompting of a plurality of potential selections associated with a process step. In other words, a process step can be a question 140, input into a window 142. The potential selections associated with the
30 question, namely the potential answers to the question, are prompted for input by a user, along with next states or steps to be followed if an answer is selected, and even next procedures if, upon selection of an answer, an entirely new procedure or process is to be accessed.

Similarly, if a procedural step is input in window 142, choices corresponding to the procedural step are input in window 154.

The "next state" section 150 includes a section
5 prompting an answer 150a, a next state 150b, and/or a next
procedure 150c. The answer is input by a user in a window
154, the next state is input to a window 156, and a next
procedure is input into a window 158. As shown by element
156, a next state can be a next sequential state (such as
10 step 3, proceeding from previous step 2) or can be a state
which skips a next sequential state or step (such as step
4 proceeding from step 2). As shown by the entry in window
158, the procedure input is typically the same procedure as
a previous state, although other procedures can be
15 accessed.

An answer most recently input by a user into window
154, along with a most recently input next state and next
procedure are highlighted and repeated in subsequent
windows 154a, 156a, and 158a. Additional potential
20 selections corresponding to an input question or procedural
step, input into window 142 for example, are also displayed
in un-highlighted form, as indicated by elements 154b,
156b, and 158b. Thus, a plurality of potential selections
are prompted and can be input by a user, associated with
25 each of the process steps or questions.

Further, although not shown, an answer or potential
section corresponding to a question or procedural step can
be a character string. In other words, the potential
selection to a procedural step may merely be a blank area,
30 requesting a user to input a string of characters. Such a
blank area would then appear on a display screen of the
created software tool, along with its corresponding
procedural step.

In addition, the Wizard builder 100 also displays

previous states in a third section of the display, as indicated by element 160, the previous states including a previous answer 160a, a previous state 160b, and a previous procedure 160c. The previous states displayed in the
5 windows corresponding to each of 160a, 160b and 160c, include respective windows 162, 164 and 166.

Finally, additional buttons are shown on the Wizard builder 100 of Fig. 3, selectable via a mouse or touch screen, for example. These buttons, a first group of which
10 is designated generally by element 170, includes a refresh button, a back button, next button, a sync main form button, and a quit button. The back and next buttons allow access to a next step or previous step, and the quit button allows the user to complete a creation of a Wizard.

15 The "refresh" button discards any user-entered data in the current state and refreshes the screen with data from the database. With regard to the "sync main form" button, this is a type of sync button for the Wizard or software tool itself. It sets the current state of the software
20 tool to the current state as shown on the Wizard builder 100 and brings the window (of the software tool) to the front of the display and makes it active. In other words, it is a way to immediately test what has been created based on entered data.

25 Additional buttons 180, 182, 184 and 186 correspond to the "answer" or potential selection section of Wizard builder 100, wherein a new answer or potential selection is added by hitting button 180; a new answer or potential selection is applied to an actual procedure or created
30 software tool by hitting button 182 (similar to applying a state of the Wizard by hitting button 122); an answer or potential selection is deleted by selecting button 184; and finally an answer or potential selection is edited by selecting button 186.

Accordingly, through the use of such buttons, and through the use of buttons for new state, applying a state, and deleting a state, a software tool cannot only be created by the Wizard builder 100, but an existing software
5 tool can be modified. For example, to add a step to an existing software tool, a stored software tool is first recalled from memory and the Wizard builder 100 then adds an input new step when the new state button 120 is activated. The state is then applied to the stored
10 procedure as a new step when button 122 is activated. Similarly, to delete a step from a stored software tool, the delete state button 124 is activated. Thus, unlike known Wizards which are preprogrammed and non-modifiable, the Wizard builder 100 of the present invention cannot only
15 easily create new software tools for any procedure or situation, but can also modify existing software tools to thereby provide for the utmost flexibility.

Similarly, not only can steps or states be added, deleted or modified, but so can answers or potential
20 selections associated with each of the process steps or state. If a new answer or potential selection is to be added, new answer button 180 is selected, and then the answer is then applied to an existing software tool by activating button 182. Similarly, an answer or potential
25 selection is deleted by activating button 184, or is modified by activating button 186. This again allows for utmost flexibility such that an existing software tool, created by Wizard builder 100, is easily modifiable to suit a user's needs.

30 Finally, a button 190 exists for generating HTML (hyper-text markup language) file(s) containing properly formatted state transition table and state table information. It is used to generate procedure information in a form suitable for review, by people such as system

engineers and procedure designers.

Note that while a display has been depicted in Fig. 3 for prompting information and for receiving information in response to prompts, such prompting and receipt of
5 information can be done by other means, such as audibly, for example.

Once information has been prompted and entered for states, state transitions, and answers, such information is assimilated to create a software tool. In the process, the
10 software tool creator may form and store state tables with corresponding attributes, and state transition tables defining movement from state to state. From this information, software tools such as that shown in Figure 4 are created, which output the input process steps based
15 upon input selection.

Figure 4 is an image of a computer screen 200 (or screenshot), displaying a software tool. created by Wizard builder 100. The display screen 200 corresponds to the exemplary question input in window 142 of the Wizard
20 builder 100 shown in Fig. 3, with such a question being displayed in a created software tool in display area 210 of Fig. 4. Additionally, the created software tool, based upon the example of Fig. 3 where two potential selections associated with the question displayed in area 210 of Fig.
25 4 have been input into window 154 of Wizard builder 100 of Fig. 3, displays two potential answers. These potential answers, displayed in the form of buttons 220a and 220b, are associated with the question displayed in area 210, and provide a user of the created software tool with two
30 choices as answers to the question posed in section 210. A mouse can be used, as can a touch screen, to select either button 220a or 220b and depending upon which one is selected, the software tool will then move to a next screen, corresponding to a next state in the process. In

the present example as shown in Figs. 3 and 4, for selection of either the duplex or simplex answer, the next state or step will be step 3.

Although not shown in Figs. 3 and 4, a choice or
5 answer corresponding to a question can be not only a preselected choice, but can be a blank window, prompting a user to input whatever information is desired. In other words, if the software tool created is that of a fax coversheet, for example, the process step shown in step 210
10 could direct a user to input name of sender. In a third area 220c (not shown in Fig. 4), a blank window could be included which, when selected, allows for a user to input his or her name. To more clearly illustrate this aspect, reference is made to Figs. 3a, 3b and 4a.

15 Fig. 3a shows a Wizard builder 100a, basically the same as but slightly modified from the Wizard builder 100 of Fig. 3. Similar to Wizard builder 100 of Fig. 3, Wizard builder 100a includes a section 108a for entry of the procedure; a section 104a for entry of the state name; a
20 section 130a listing all states in a procedure, highlighting the current state in area 132a, etc. The Wizard builder 100a of Fig. 3a further illustrates a window 141 which permits selection of a question type. As shown in window 141 of Fig. 3a, one such question type is a
25 "string". Selection of "string" as a question type creates a software tool which prompts a user to input a character string in response to the question posed. Such a question is input into window 142a. The range of the string permitted is then input into windows 156a (minimum) and
30 158a (maximum). A software tool, as shown in Fig. 4a, is then created from which the input process steps are executed based upon input selections.

Figure 4a is an image of a computer screen (or screenshot), displaying a created software tool. As shown

by element 210a of Fig. 4a, in the created software tool, the question is displayed. In the present example, the question "What is your name?" was input into window 142a of the Wizard builder 100a of Fig. 3a, and that question is
5 displayed in the created software tool. A blank window 225 of Fig. 4 is then displayed, prompting a user to input a string of characters (numbers, letters, etc.) as a variable answer to the question. In the example shown in Fig. 4a, the name "Don" has been entered by the user. Thereafter,
10 since the default "OK" is shown in the answer window 154a of Wizard builder 100a of Fig. 3a in the present example, an "OK" button 226 is displayed to the user in the screenshot of the created software tool of Fig. 4a. The user then hits this button after the character string has
15 been entered in window 225 to move to step 2 of the procedure.

Fig. 3b illustrates another Wizard builder 100b, basically the same but again slightly modified from the Wizard builder 100 of Fig. 3. Again, many similar windows
20 are shown, which will not be discussed further for the sake of brevity. The Wizard builder 100b illustrates that not only can a selectable question type include "string", but other question types are also selectable via drop-down selection box window 141a. These include, but are not
25 limited to: lists, ranges, strings, access IDs, object IDs, IP addresses, dates, times, etc. Thus, as basically shown in Fig. 4a, when the created software tool encounters any one of the selected question types, it displays the entered question, a blank window (such as a blank text-box
30 or character box) in which a user of the created software tool can type an answer, and an "OK" button to hit when the answer has been entered (optionally, for example, a user could hit an "enter" key on a keyboard instead of the "OK" button), or a list of allowed answers.

Additionally, the Wizard builder 100a of Fig. 3a also includes a window 146 labeled "Answer (i.e., Variable) Name". This allows one to name a variable in which the created software tool can store a user's answer. For example, such an aspect can be optional for list-type questions and mandatory for string-type questions. Thus, an input string answer will be stored. These answers can further be used in other questions and in machine-interface code.

10 For questions, the simple way to do this is to prepend a marker (such as a dollar sign "\$"). Thus, if question one is "What is your name?", and an answer of "Don" is entered, then a subsequent question may say "OK Don, what do you want to do now?"

15 As shown in Fig 4, buttons such as buttons 230a and b, and 240a to c are typically displayed in a created software tool. These buttons can be selected by a user, via a touch screen, or can be selected through the use of a mouse or keyboard for example. The restart button, 230a, restarts
20 the entire software tool at the first state or step, displaying the first process step. The "back" button 230b, allows the user to move to a previous step in a procedure. The "more-info" button 240a and help button 240b, provide a user further information corresponding to a process step
25 or selectable answer, or provide a user with additional information to aid them in answering a particular question, respectively. For example, "More Info" could bring up a Web browser or fancy display of some kind, while "Help" might bring up the standard help aspect of the operating
30 system. The button 240c allows the user to quit using a particular software tool created by the Wizard builder 100 of Fig. 3.

Finally, as shown in Fig. 4, the procedure corresponding that input in window 108 of Fig. 3, is

displayed as element 250. Also, the step input in window 104 of Fig. 3, is displayed as element 260. This lets a user know which procedure and step are being executed.

As previously indicated, the interface 36 of the Wizard 200 can be visual or audible. Similarly, audible output or visual output are not limited to the English language. For example, in a section 270 shown in Fig. 4, one of a plurality of languages is selected from an area 280, such as English designated by 282. Using an arrow in area 280, one chooses from a plurality of predetermined languages (from a pull-down window for example) in which to display or audibly output the process steps and potential selections associated with the process steps for the software tool, such as English, French, Spanish, etc. Thus, the process steps and potential selections associated with the process steps in the generated software tool can be audibly output or displayed in any one of a plurality of languages.

Fig. 5 is an example of a sample state table 300, listing a plurality of states or steps in a procedure, and corresponding attributes. This state table 300 is compiled from prompted and input state and attribute information and stored in memory 34, for example. Information from the state table 300 is then used to generate a software tool.

The state table includes, for example, a first column 310, corresponding to the overall procedure or method. As shown, each of the five rows a-e of state table 300 all correspond to the same procedure, namely procedure 1. The second column of the state table 320, indicates a step in the procedure, such as step 2, step 3, etc. A third column 330, corresponds to an attribute of the step, indicating whether or not a step is a default step. In this example, only row d, the start step, is such a default step. The next column 340 indicates the text of the step, namely the

question or process corresponding to the step.

The fifth column 350 of state table 300 corresponds to another attribute indicating the step type, namely human or machine. The sixth column 360 corresponds to an optional
5 attribute, a URL address, wherein row d is the only step with a URL address. The final column 370 corresponds optionally to a terminal (ie, ending) state. Accordingly, input procedures, states, questions, etc., may be stored in a state table such as that of Fig. 5 for example, or
10 information prestored in such a format may be used, to create a software tool using Wizard builder 100. Similarly, information stored in a state transition table such as that shown in Fig. 6, or information prestored in a state transition table may be used.

15 The state transition table shown in Fig. 6, listing movements from state to state in response to a given input, is stored in memory 34, for example. It can be created from input state transition information (or prestored) and used in conjunction with the state table shown in Fig. 5 in
20 the creation of a software tool utilizing Wizard builder 100 of Fig. 3 for example. The state transition table 400 includes, for example, a first column 410, indicating a "from state" or current step in a procedure; a second column 420 corresponds to an event or answer to the current
25 question or step; and column 430 corresponds to a "to state" or a "next" state in a procedure.

Accordingly, as shown in row a of Fig. 6, for example, if a current state is start, and either the event "maybe" or "yes" is selected, then the created software tool moves
30 to the "to state" or "next state", namely to step 2 in the process. Alternatively, if the event (answer) selected after the start step is no, the created tool moves to step 3 in the process (as shown in row d for example). Additionally, a URL address can optionally be input in a

fourth column 440, if necessary. Additionally, it should be noted that the state table and state transition table can be stored in a relational data base, from which such a software tool is created, such as that described in
5 copending application serial nos. 09/182,637 and 09/182,493, respectively entitled "Method and Apparatus for Storing Data as Liquid Information" and "Method and Apparatus for Amplifying Design Information into Software Products", the entire contents of which are incorporated by
10 reference.

The Wizard builder 100, 100a and 100b of the present invention allows a user to read, enter, and edit information, such as information in the form of a state table and a state transition table for example, in an
15 efficient way to create an effective software tool. Such tables are preferentially stored in a relational database.

To achieve the desired result, all states are listed in window 130 of Figs. 3, 3a and 3b, and a user chooses one. Choosing one fills windows in the top half of the
20 screen (above window 150) of the Wizard builder, displaying important information about the state. The top half of the screen contains windows and controls that display and allow editing of attributes of the current state. Since there may be more information than can fit on one screen, an edit
25 state button can be used to bring up another screen or screens to enter the lesser-used state information.

The top part of the bottom half of the screen (from window 150 and below) of the Wizard builder of Figs. 3, 3a and 3b allows a user to read, enter, and edit state
30 transition table information (answer information), referring to possible answers to the current question and their next states. An Edit Answer button can then be used to bring up additional screens if necessary. Finally, the bottom part of the bottom half of the screen provides

controls for navigation and miscellaneous functions, such as navigation aids for moving among states and provides miscellaneous functions (such as generate HTML).

Hereafter, an example of a procedure implemented into a software tool by Wizard builder 100 will be discussed, along with display screens (screenshots) of the resulting software tool. The Wizard builder 100 of Fig. 3 is utilized to create a software tool from the flowchart of Fig. 7, wherein the various screenshots of the software tool are shown in Figs. 8 to 11.

Initially, in creating a software tool from the Wizard builder 100 of Fig. 3 of the present invention, a flowchart such as that shown in Fig. 7 can be, but not need be, created. Any process or procedure can be used to create a software tool using Wizard building 100. The present example is illustrated in flowchart form for illustrative purposes only.

The first step of Fig. 7, step S1, is a procedural step stating "open cabinet door". In step S2, it is questioned whether or not "ok" has been selected (indicating that a procedural step has been completed) and if so, the process proceeds to step S3.

Fig. 8 is a screenshot 400a of a software tool created by Wizard builder 100 of Fig. 3, implementing procedural steps S1 and S2 of Fig. 7. In essence, steps S1 and S2 of Fig. 7 involve a single state or step (step 1) and a possible selection corresponding to the step (noting that there is only one possible selection in the current example), which are input into Wizard builder 100. Step 1 required a human response, and thus when creating the software tool resulting in the screenshot of Fig. 8, "H" was input in window 112 of Wizard builder 100 of Fig. 3. The procedure, as displayed in element 450A of Fig. 8, is "PROC 1", which was input into window 108 of Wizard builder

100, and the step S1 is displayed as element 460 in Fig. 8. The question or process step input into window 142 of Wizard builder 100 of Fig. 3, included step S1, namely "Open cabinet door" and included the further instructions
5 "hit ok when done," and is displayed as element 410a in the created software tool.

Finally, since the only selectable potential selection is "ok", based upon the fact that the flowchart of Fig. 4 is only an example of a simple sequential procedure (with
10 no conditional steps involving multiple choices), a single answer was input in window 154 of Wizard builder 100 of Fig. 3, and is displayed as a display button 420a of Fig. 8. The next state input in window 156 of Fig. 3 of Wizard builder 100 was "2", and the next procedure was procedure
15 PROC 1, input into window 158 of Wizard builder 100 of Fig. 3.

Fig. 9 is a screen shot 400b of the software tool, for steps S3 and S4 of the procedure, somewhat similar to that of screenshot 400a for steps S1 and S2 shown in Fig. 8.
20 The second step of the software tool includes both step S3 and S4 displayed in area 410b, displaying "Replace board labeled ACU with new one. Hit ok when done." Again, the only possible potential selection is "ok", and this is displayed in area 420b of Fig. 9. The procedure remains
25 PROC 1 as displayed in area 450b, and the step is now Step 2 as displayed in area 460b.

Somewhat similarly, in Fig. 10 in a screenshot 400c of the software tool, for steps S5 and S6 of the procedure. The steps are displayed in area 410c as "Close cabinet
30 door. Hit ok when done." Again, the only potential selection is "ok", and this is displayed as a single button 420c. The procedure remains PROC 1 as displayed in area 450c, and the step is now Step 3 as displayed in area 460c.

Finally, Fig. 11 is a screenshot 400d of the software

tool, for the last step S7 of Fig. 7. The last step 410d is displayed as "You are done with this procedure." No potential selections or answers are displayed, since this is the last step of the procedure. The procedure remains
5 PROC 1 as displayed in area 450d, and the step moves to Step 4 as shown in area 460d.

Accordingly, as shown in Figs. 8 to 11, the Wizard builder 100 of Fig. 3 is utilized to create a software tool which prompts a user through four display screens, to lead
10 them through a step-by-step procedure. The example shown in Figs. 7 through 11 is a procedure for repairing a problem in a telecommunication system and illustrates how a complex procedure can be simplified by creating a software tool (Wizard) utilizing the Wizard builder 100 of
15 the present invention.

A second example will be discussed, utilizing Figs. 12 to 18, indicating conditional steps or multiple potential selections corresponding to a single process step. It should be noted that real-world examples of software tools
20 which can be created by Wizard builder 100 may be far more complex than those shown in either of Figs. 7 to 11 or 12 to 18.

The procedure to be implemented in a form of a software tool (Wizard) is shown in Fig. 12 and includes
25 steps S10, S12, S14, S16, S18, S20, S22 and S24. Fig. 13 is a first screenshot 500a of a software tool created using a Wizard builder 100 of the present invention. The first step in the procedure of Fig. 12 is displayed in area 510a, which essentially encompasses steps S10 and S12 of Fig. 12.
30 Area 510a displays the process step "Open cabinet door. Hit ok when done." Since there is only one potential selection corresponding to the procedure step, this potential selection is displayed as an "ok" button in area 520a. The procedure is identified as procedure 2 in area

550a and the step is indicated as step 1 in area 560a.

Fig. 14 is a screenshot 500b of the software tool. It illustrates a screenshot with a conditional procedural step. Namely, this corresponds to step S14 of Fig. 12, and
5 the question is displayed in an area 510b as "Is fault LED lit on board labelled ACU?" Instead of just displaying a single potential selection associated with the process step, areas 520b and 525b display two different potential selections, namely a first selection "no" in an area 520b
10 and a second selection "yes" in area 525b. The procedure "PROC 2" is displayed in area 550b and the step, step 2, is displayed in area 560b.

To further illustrate the conditional step of the software tool screenshot of Fig. 14, Fig. 15 is presented.
15 The Wizard builder 600 of Fig. 15 is essentially the same as Wizard builder 100 of Fig. 3, with the exception that the steps, procedures, questions and answers are modified so as to correspond with various steps shown in Fig. 12 and displayed in Fig. 14. As shown in area 630 of Wizard
20 builder 600, the software tool implementing the procedure of the flowchart of Fig. 12 is a 5-step procedure, with step 2 being highlighted in area 632 of Fig. 15. This step 2, corresponding to the screenshot of Fig. 14, is the state name shown by step 2 in window 604. The procedure PROC 2
25 is input into window 608.

As shown by the process step displayed in portion 510b of Fig. 14, the question is initially input into the Wizard builder 600 in area 642. In order to create the two potential selections shown in Fig. 14, the two selections
30 are each input in the "answer" section of the Wizard builder 600 of Fig. 15. For example, the answer or potential selection "yes" is first input as shown in area 654b, with its corresponding next state "step 3" in area 656b, and its corresponding procedure "PROC 2" in area

658b. The second potential answer is input into area 654, and displayed as "no" in area 654a; its corresponding next state is input in area 656 as "step 4", and displayed in area 656a; and its procedure is input in window 658 as
5 "procedure 2", and is displayed in area 658a.

Finally, in the previous "states" section of Wizard builder 600, the previous answer "ok" is displayed in area 662; the previous state "step 1" is displayed in area 664; and the previous procedure "PROC 2" is displayed in area
10 666. Thus, Fig. 15 illustrates Wizard builder 600, in which the conditional step of step S14, and the conditional "yes" and "no" answers leading from step 14 to different next states, are first input. This then results in the creation of the software tool and the display of screenshot
15 500b as shown in Fig. 14.

Moving back to the software tool created based upon the flowchart of Fig. 12, Fig. 16 is a screenshot 550c, for step 3 of the software tool procedure, reflecting steps S16 and S18 of Fig. 12. The step, "step 3", is displayed in
20 area 560c, and is only encountered when the answer to step S14 of Fig. 12 is "yes", and when the "yes" button 525b of the screenshot 500b is selected. The procedure remains PROC 2 as displayed in area 550c, and the process step of essentially steps S16 and S18 are displayed in area 510c,
25 namely "Replace board labelled ACU with new one. Hit ok when done." Since the only potential selection associated with the process step is "ok", an "ok" button is displayed in area 520c.

Again, when viewing Figs. 14 and 16, one can see that
30 the software tool created from the flowchart of Fig. 12 easily enables a user to step through a complex procedure such that when a fault LED is lit on the board labelled ACU, the board labelled ACU will be replaced with a new one based upon a procedure moving from step 2 of the screenshot

of Fig. 14 to step 3 of the screenshot of Fig. 16. To the contrary, if "no" is selected in the step of the software tool shown in the screenshot of Fig. 14, then the software tool will not display the screenshot shown in Fig. 16, but
5 will instead display the screenshot of Fig. 17.

Fig. 17 is a screenshot 500d which, in essence, represents steps S20 and S22 of Fig. 12, and is essentially step 4 in a 5-step software tool procedure. Step 4 is displayed in area 560d, with the procedure remaining PROC
10 2, as displayed in area 550d. The screenshot 500d of Fig. 17 displays the process step of steps S20 and S22 as "Close cabinet door. Hit ok when done." in area 510d, and displays a single potential selection "ok" in area 520d.

Thereafter, the process proceeds to step S24, which is
15 reflected in the fifth screenshot 500e of the created software tool shown in Fig. 18. The phrase "You are done with this procedure." is displayed in area 510e. Since this is the end of the procedure, no potential selections are displayed. The procedure is again displayed as PROC 2
20 in area 550e and the step, as indicated by step 5, is indicated in 560e.

Accordingly, in this second example shown in Figs. 14 to 18, a 5-step software tool is created by the method and apparatus of the present invention, from the conditional
25 procedure illustrated by the flowchart of Fig. 12. Similar to the first example given, the procedure of Fig. 12 is basically a step-by-step procedure, demonstrating a multiple choice answer wherein, depending upon the selection made by the user, a new screenshot of the created
30 software tool is displayed.

Accordingly, with the Wizard builder of the present invention, a user is prompted to input process steps of a procedure, and a plurality of potential selections, either visually or audibly for example. A state or step of the

procedure is essentially a state of a finite state machine, and corresponds to a step of a procedure or question of either a human end-user, of another system or machine, or of another program of a machine. The Wizard builder 100, 5 Fig. 3 prompts input of process steps, potential selections associated with each of the process steps; stores the input process steps and associated potential selections; and creates a software tool, such as that displayed in Figs. 13, 14 and 16-18.

10 Additionally, next states can be prompted, or automatically generated if only a single potential selection corresponds to an input process step. A URL address (world-wide-web uniform resource locator) may be used to provide additional information (such as "help", for 15 example) and can be input into the Wizard builder. The Wizard builder can be used to designate a state as a starting state, an ending state, and also can prompt next state information or retrieve such information from a prestored state transition table.

20 As in any finite state machine, a set of state transitions, specifying how a machine moves from state-to-state in response to inputs, is prompted and input to the Wizard builder 100 of Fig. 3 of the present invention. Further, the Wizard builder of the present invention allows 25 for not only the creation of a software tool, but also the creation of a software tool which is easily modifiable by the user wherein states can be added, applied and deleted and answers or potential selections corresponding to a state can be added, applied, deleted or edited.

30 In addition, the present invention should not be limited to an apparatus implemented by a CPU, memory and interface as shown in Figs. 2 and 3 of the present invention. The invention may also be implemented in other forms, such as an article of manufacture as shown in Fig.

19, for example. Essentially, the entire Wizard builder 100 of the present invention can be implemented in the form of an article of manufacture, 700, as shown in Fig. 19 such as a computer disk, optical disk, CD ROM, or other medium storing the information necessary to create the Wizard builder 100 of Fig. 3. The article of manufacture 700 preferably comprises a computer usable medium such as a memory 710 (which may be a computer disk, RAM, ROM, optical disk, etc.) including a first code 720A for causing a computer (such as CPU 32 for example) to prompt input of process steps and a plurality of potential selections associated with each of the process steps; a second code 720B for causing a computer to store input process steps and associated potential selections; and a third code 720C for causing a computer to create a software tool based upon the stored process steps and associated potential selections. The various codes cause the computer to prompt a displayed template such as that shown in Fig. 3 of the present application corresponding to a Wizard builder 100, or cause the computer to audibly prompt information, for example.

Alternatively, as shown in Fig. 20, the Wizard builder 100 can be implemented in the form of a propagated signal 800, transmitted from one source 810 to another source 820, for example. The first source 810, sends the propagated signal, including a first code segment instructing prompting input of process steps in a plurality of potential selections associated with each of the process steps; a second code segment instructing storage of input process steps and associated potential selections; and a third code segment instructing creation of a software tool based upon the stored process steps in associated potential selections. The signal 800 containing such information is then propagated over the world-wide web, through air-waves,

etc., to a second location 820, including a computer (such as CPU 32, for example) for receiving the propagated signal, for example. Such a propagated signal is one embodied in a digital signal, a digital bit-stream, or a
5 carrier wave for example. The code segments instruct prompting via a display template, audible prompting, etc.

The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and
10 scope of the invention, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.

5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995